AI CUBED HOLIDAY PROGRAM '22

INTRODUCTION TO THE METAVERSE

DRAGON REALM VR

PART I

Designed and prepared by Dion Stojsavljevic

Welcome adventurer, to the world of Dragon Realm!

Goal of this course

Become familiar with HTML and A-Frame to build a VR world using a step-by-step guide, and then solve challenges by adding your own custom elements to expand the world.

The goal is to get you to a level of competence with A-Frame so you can research on your own, beyond the content of this course, and expand your Dragon Realm VR world.

What you will learn

As you progress through this workbook, you will learn the fundamentals of VR design in A-Frame. You will build a quick reference guide of all the key terms you need to build VR worlds, including:

Code layout best practice	Importing rendered 3D elements
Plotting coordinates	Creating skylines
Basic shapes – boxes, spheres, cones, cylinders	Click and gaze based interactions
Changing properties: colour, position, size, texture	Adding environmental elements – mood lighting and rain
Animation	Modifying the camera and making paths
Fundamentals of good VR design	Viewing your world on a VR headset

How to use this guide

This guide is designed to be read in order, to help you build your own Dragon Realm VR world.

You will find a blank reference guide template in the appendix of this document, that you can print and fill out as you learn new code and concepts, so you have all the key terms handy as you work.

It is very important you READ all the instructions on every page or you may not be able to solve some of the questions!

1. Welcome to Dragon Realm!

In this project you will build a VR world set in medieval times, with the center of your world a stone castle, complete with a lowering drawbridge.

Along the way you can choose the colors you want to use, and through challenges, choose what additional elements you would like to add to your world, like:

- lighting and weather
- dragons
- fairies
- furniture
- flora and fauna
- landscapes and more.

Everyone will end this course with an animated scene that looks similar to the image below, that you can navigate with your own VR headset. For some of you who work quickly or are familiar with HTML or A-Frame already, there is more challenging content as your progress.

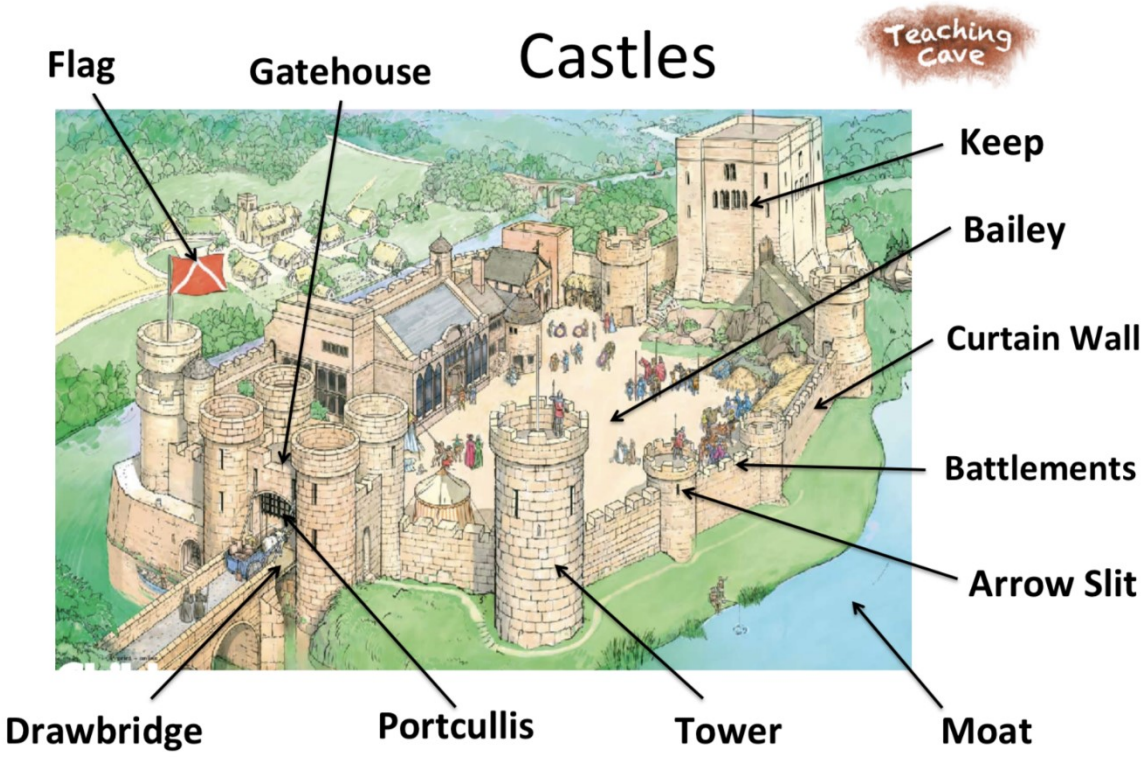
Lets get started!

1.1 It all starts with a castle….

Wikipedia defines a castle as the following:

A castle is a type of fortified structure built during the Middle Ages predominantly by the nobility or royalty and by military orders. Scholars debate the scope of the word castle, but usually consider it to be the private fortified residence of a lord or noble.

The image below (source: https://www.teachingcave.com/world-around-us/ks1/castles/) shows all the key elements found in a typical castle, most of which we will include in our design.



Your castle, your style..

As you build, there will be elements that you will need to make exactly as instructed. There will be opportunity to add your own style to your castle and world based on your imagination.

Write a short story on the following blank page about who will be living at your castle and a narrative of the world they live in. This will help you decide what elements you would like to add as you go.

1.2 My Dragon Realm Story

Remember to include WHO lives in your castle, who they are defending themselves from (or is it a peaceful world?), what type of environment it is (i.e., desert, forest, snow, mountains, alien!) and the weather (i.e., day/night, hot or cold, raining or sunny)

Write a list below of all the elements you would like your castle to include, and any special items like furniture etc.

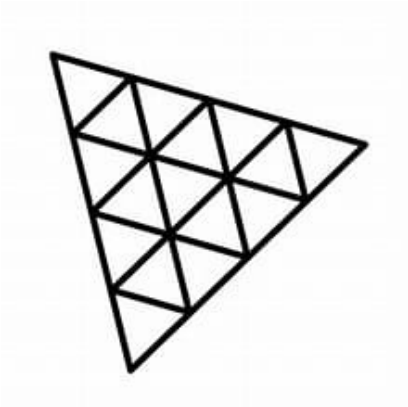
2. Getting started

Brief overview of HTML and A-Frame (assumption is most students will be familiar with this already)

What is A-Frame?

A-Frame is an open-source web framework for building virtual reality (VR) experiences. It is maintained by developers from Supermedium (Diego Marcos, Kevin Ngo) and Google (Don McCurdy). A-Frame is an entity component system framework for Three.js where developers can create 3D and WebVR scenes using HTML. HTML provides a familiar authoring tool for web developers and designers while incorporating a popular game development pattern used by engines such as Unity. *Source: Wikipedia*.

What is Three.js?



Three.js is a cross-browser JavaScript library and application programming interface (API) used to create and display animated 3D computer graphics in a web browser using WebGL. The source code is hosted in a repository on GitHub.

Source: Wikipedia.

What do I use to build my Dragon Realm?

We will be using a very popular tool, used in industry, called **Microsoft Visual Studio Code.**

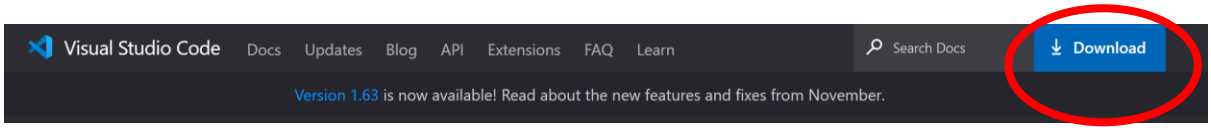
Visual Studio Code is a source-code editor made by Microsoft for Windows, Linux and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git. Users can change the theme, keyboard shortcuts, preferences, and install extensions that add additional functionality.

Source: Microsoft

2.1 Setting up MS Visual studio code

Go to: https://code.visualstudio.com/

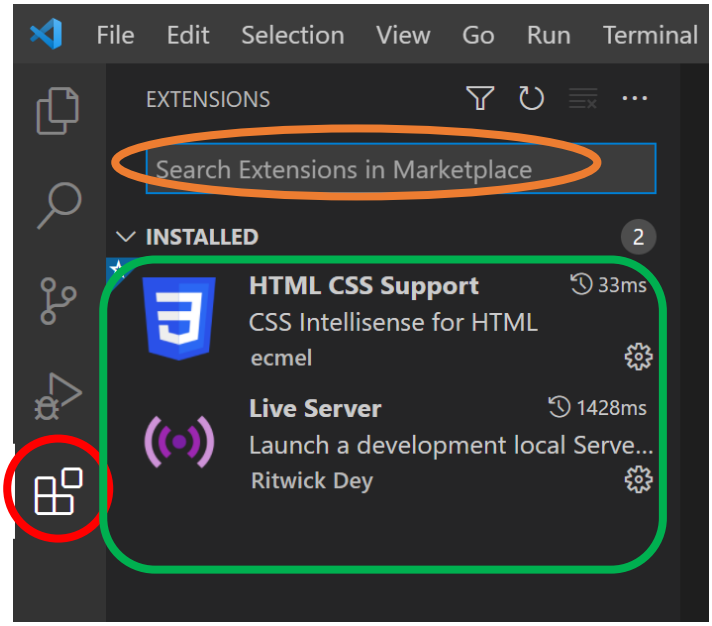
Click ***'Download'*** at the top right-hand corner of the screen:



Once your download has finished, run the setup program and follow the onscreen instructions.

*Note: During setup, if it asks you what coding environment you want to optimize for**, select HTML.***

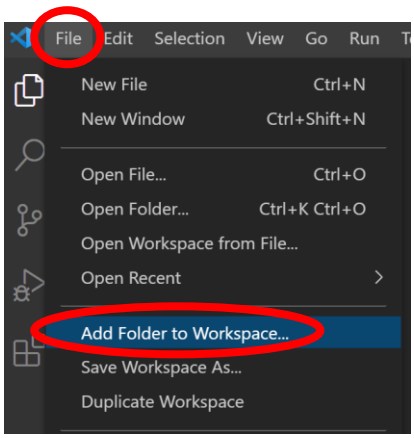
- Once it is installed, open the program, and select the ***EXTENSIONS*** **icon**
- In the **SEARCH BAR** type ***live server*** and install the application
- Then type ***HTML CSS Support*** and install that application also
- When completed, you should have **both apps show in the list** like shown on the right (3)



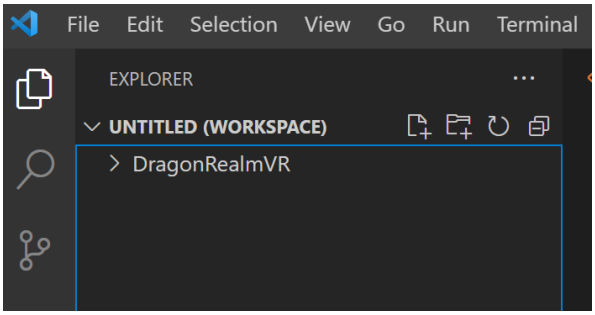
2.2 Setting up your workspace

2.3

- Click file in the menu, and select Add Folder to Workspace…
- Choose a **location and a name your folder DragonRealmVR**.
- Note: It is strongly recommended not to put it in a deep directory structure, and just put it somewhere simple like on your desktop. Ask your teacher if you are unsure what this means.



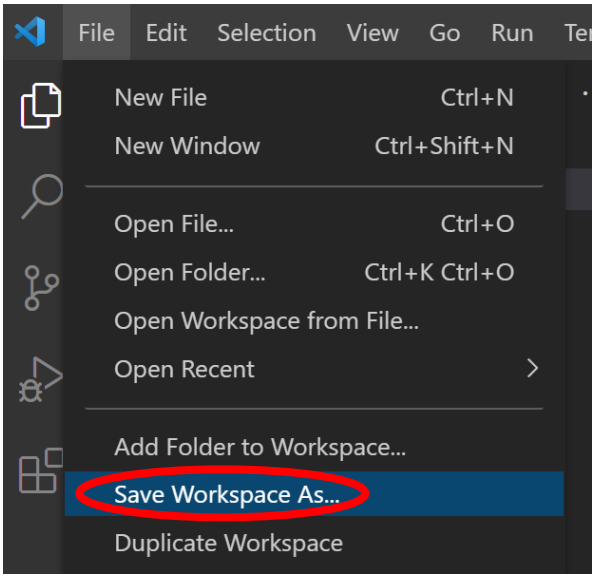
Your folder should appear like shown on the right



Next, save your workspace.

You will need to select the same folder that you just created (DragonRealmVR) and save the workspace with the name Dragon Realm VR

Now you are ready to download the blank template file.

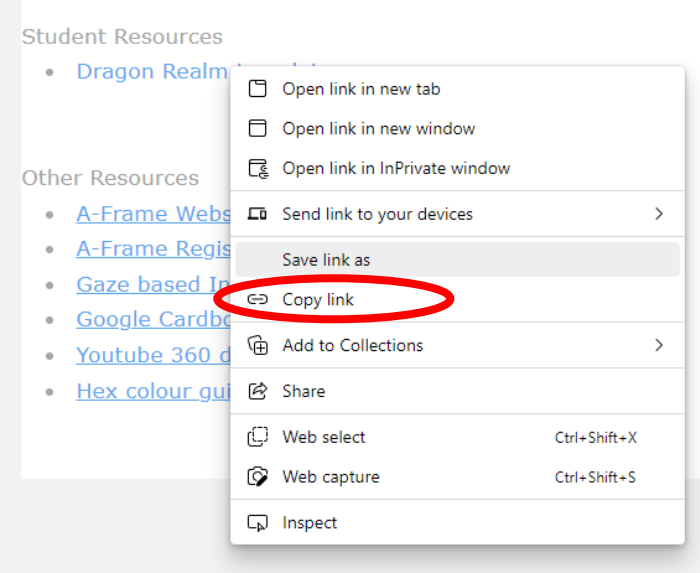


- *Go to your exclusive resource website at:* www.ai3.academy/inspire/metaverse

- Under the student resources heading, RIGHT CLICK the file 'Dragon Realm template' to open the menu - ***DO NOT LEFT CLICK THE LINK!***

- Select ***save link as*** and then choose the folder you made previously (DragonRealmVR)

- Your HTML template will now be saved in your workspace, and you are ready to start!



2.3 HTML and A-FRAME Formatting Basics

Because A-Frame is based on HTML, it uses opening and closing tags the same way as HTML. Let's do a quick recap on some HTML basics to keep your code formatted in the best way.

Using Tags

- You always start your code with an 'open' tag like this:
  -
  - The above example is the command to turn text into bold font.
- And a closing tag would look like this:
  -
  - Note that there is a backslash after the first bracket to signify the tag can end.
- A full usage of this tag would look like this:
  - All this text will now be bold
- And the result would look like this:
  - **All this text will now be bold**

Layout and nesting

When you have a tag within a tag, it's important to visually be able to see the relationship between them. This makes it easy to see where one starts and ends.

Have a look at the section below and note the following:

- Line 96: This is a comment and is only used to make notes inside your code. This is important for you to be able to remember code you created and find it later, or for others to do the same.
- Line 97: Note that this is the starting tag, and the closing tag is in the same column on line 108.
- Note how each tag within the section is indented by one tab stop. This makes it much easier to debug or make changes to later.

```
96      <!-- Four -->
97          <section id="four" class="wrapper style3">
98              <div class="inner">
99
100                 <header class="align-center">
101
102                     <h3>“Education is not the learning of facts, but the training of the mind to think.”
103                     </h3>
104
105                     <p>Albert Einstein</p>
106                 </header>
107             </div>
108         </section>
```

2.4 Exploring VR_CASTLE_TEMPLATE.html

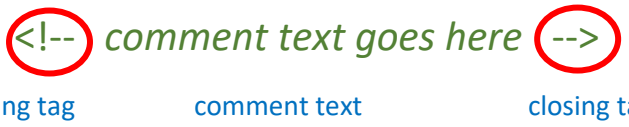
Now that you have Visual Studio Code setup, let's explore the template file that you downloaded.

Click on the ***VR_CASTLE_TEMPLATE.html*** file in your workspace.

The top of the file should look like the image below.

Once you have looked over the code, have a look at the following items:

- Lines 6 and 10 – Both of these lines have text displayed as a comment, and they won't be run in your program. The comment format is as follows:



- There is a difference between the lines however. Line 6 is a heading to let you know what that section contains. Line 10 is a line of code that has been 'commented out' for later use.

It's important you don't change or delete these comments as you will be asked to use them later on.

Using A-Frame JavaScript libraries

Remember that A-Frame is a platform that uses HTML, and references pre-made JavaScript libraries of code.

- The A-Frame tags point to and run JavaScript code in reference libraries.
- For this project, we will be using the three libraries you can see on lines 7, 8 and 9 of your template.
- This is important as some different A-Frame tags and features may only work with certain versions of the JavaScript library.

The three we have pre-loaded for you will all work with the instructions in this guide unless indicated otherwise.

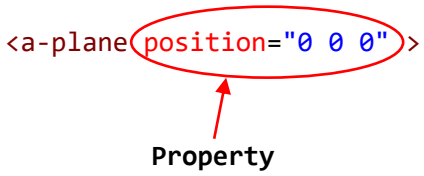
2.5 Building the Ground for your world

First we need to build the ground, or 'floor' of our world.

1. Find the following comment: <!-- Floor -->
2. Under the comment create the opening tag: <a-plane>
3. On the line below create a closing tag: </a-plane>

We now have the tags in place, but need to add some information (position, size and color) about what our floor will look like. This information is called the tag PROPERTIES.

The properties go just before the end bracket of the opening tag, like this:

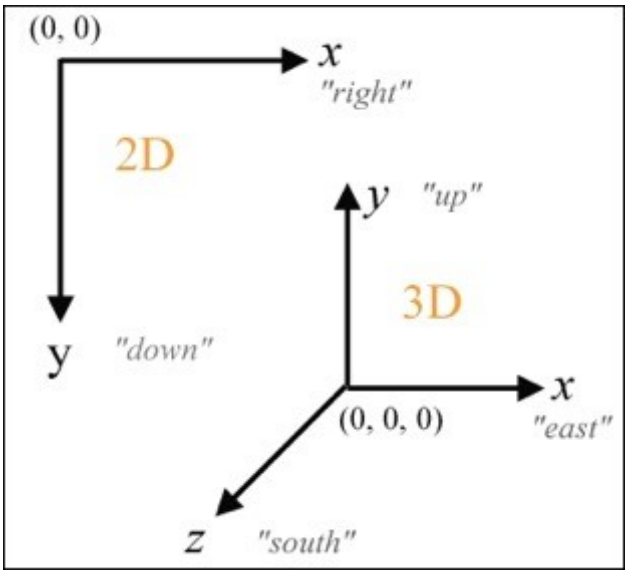


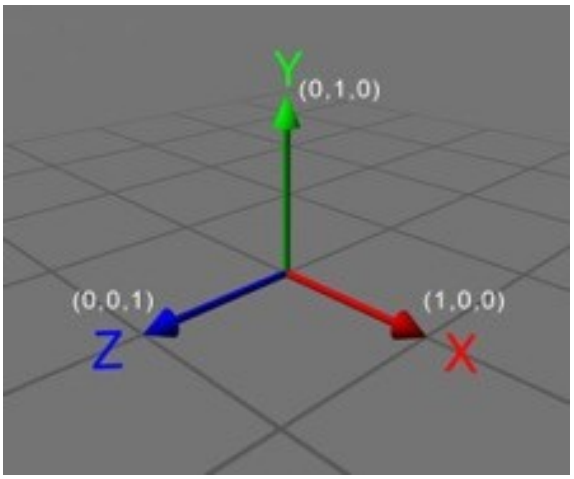
In this case the property is setting the position we want the floor to appear. The three numbers are, in order, the X, Y and Z coordinates.

Introduction to coordinates

Unlike 2d design which uses two axis (X and Y), 3d design uses 3 axis's for understanding the position of our 3d models (X, Y and Z).

- **X** referring to the horizontal (Left to right)
- **Y** referring to vertical (Up and down)
- **Z** referring to how far forward or back our 3d design is in our workspace.



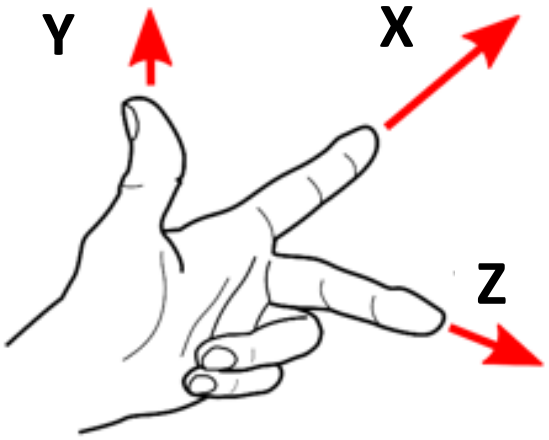


This image shows which number relates to which coordinate. They are always written in the same order, no matter what programming language you are using: X then Y and Z last.

A 'handy' way to remember these when you are working is to use this system with your left hand.

The direction each finger points is the positive (+) or forward direction for each coordinate.

0,0,0 is always the centre of the screen.



- Now that we know a bit more about coordinates, lets get back to building our floor…. This is what we have so far:

```
<a-plane position="0 0 0">
```

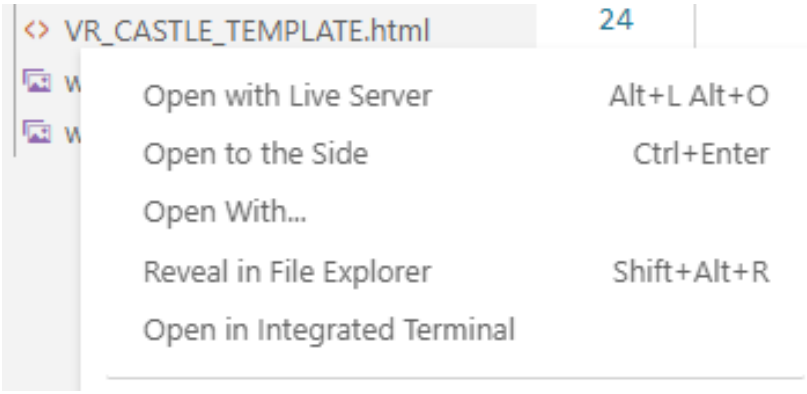
- Now we are going to add properties for the **size and color** *(note that we need to use American spelling for color!)* of our floor:

```
<a-plane position="0 0 0" width="100" height="100" color="green">
```

Time to run our code and see what happens!

To run your code:

Right click on the template HTML file, and then select ***Open with Live Server***



If it works, you will realise one of two things:

1) You have fallen over and are suddenly looking at your screen while lying on your side!
2) The floor is not in the correct position and is running vertically up the screen!

I will assume you are still sitting comfortably in your chair, so it will be the second option! Let's fix it….

- Let's add a ROTATION property to our floor.

```
<a-plane position="0 0 0" rotation="-90 0 0" width="100" height="100"
                                color="green">
```

Now run it again and see what happens!

The rotation property uses the same X, Y, and Z coordinates as position. In our example, we rotated the X axis minus 90 degrees so it would appear horizontal.

- Experiment changing the rotation, size and color properties of your new floor to understand how each value works. Once you are satisfied how they work, set it back to the example above and we are ready to start our castle!

Note: We will build using basic shapes first and add the detail and 'flair' at a later point, so don't worry if it doesn't look like the example yet!

3. Building the Castle

In this section we will build the main castle structure. Here are all the elements we will need to construct. This should take between one to two full days to complete

Front left tower – learning basic shapes

1. tower - cylinder
2. window - box
3. roof - cone

Front wall

1. Wall above draw bridge - Four boxes
2. Battlements on top of wall
3. Front wall left of drawbridge
4. Front wall on right of drawbridge

Other Castle towers – Use a Template and maths to work out locations

1. Front right
2. Back left
3. Back right
4. Middle left
5. Middle right

Rest of castle walls – Use a Template and maths to work out locations

1. Rear wall
2. Left wall back
3. Left wall font
4. Right wall back
5. Right wall front

Challenges

1. Add a castle Keep
2. Wall battlements
3. Fortify one of the towers

Adding finishing touches

1. Adding surface textures
2. Building a sky
3. Flags on towers
4. Castle Floor
5. Moat – Ring

3.1 Front left tower – learning basic shapes

The main tower will use a CYLINDER. Here is the Tags we will use:

```
<a-cylinder> </a-cylinder>
```

Using the same PROPERTY format as the Floor we just created, add the following properties:

- `Red Color`
- `Position 0 0 0`
- `Width of 2`
- `Height of 2`
- `Depth of 2`

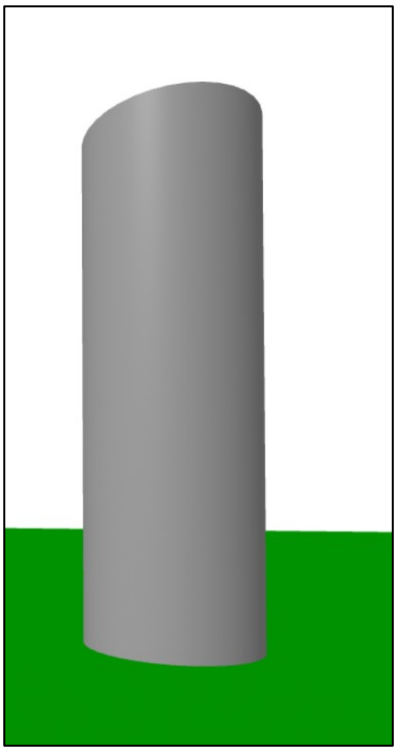
Run your code and have a look at the result. Can you see the red cylinder?

We want to make a couple changes to start it further away from us, and also to make it look more like a castle tower.

Let's update as follows and see the differences:

- `Position -2 3 -6`
- `Height of 6`
- `Grey Color`

Our result should look like the image to the right.



3.2 Tower Window

Now lets add a window using a box tag:

```
<a-box> </a-box>
```

- `Black color`
- `Position -2 4.75 -5`
- `Depth 0.1`
- `Set your own width and height.`

Hint: have a look at page one and two of this guide to help with your window sizing.

3.3 Tower Roof

For our tower roof, we will add our third shape, a cone. A cone needs slightly more information as we need to set the radius of the top and bottom.

Here is the cone tag:

```
<a-cone> </a-cone>
```

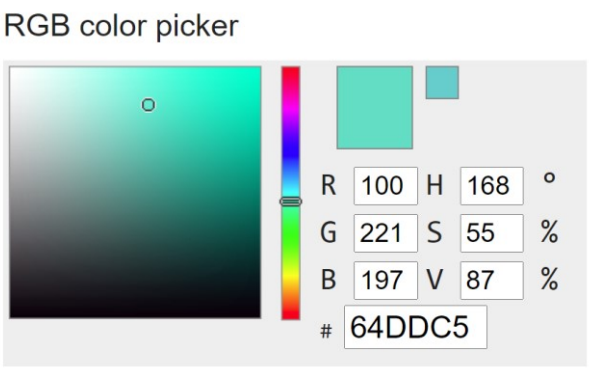
Now lets set the properties. As there are some new properties here, you can just copy these exactly as they are:

- `color="#64ddc5"`
- `radius-bottom="1.2"`
- `radius-top="0.4"`
- `height="1"`
- `position="-2 6.5 -6"`

You will notice we have used a code for the color. This is called a Hex code, and it's a powerful way to select from a LOT of colors….
Here is a great resource you can try to generate your own Hex codes:
https://www.rapidtables.com/web/color/RGB_Color.html

You will see from the example on the right, there are two different color codes we can generate.

- This one is showing our example ="#64ddc5
- Note you will need to add the hash tag in front for it to work in your code.
- It can also be represented as a RGB color, with the code rgb(100, 221, 197)



For each colour (Reg, Green and Blue) they have 256 different tones, from 0 to 255.

You can work out how many colors you can generate by multiplying all 3 combinations together:

256 x 256 x 256 = ☐ 😮 *Yes, it's a lot!*

For our final touch on the tower, we will add a flag on a pole. Later on, you can choose your own picture to put on the flag or change the shape.

3.4 Flag

Let's start with a flag pole first, using a cylinder with the following properties:

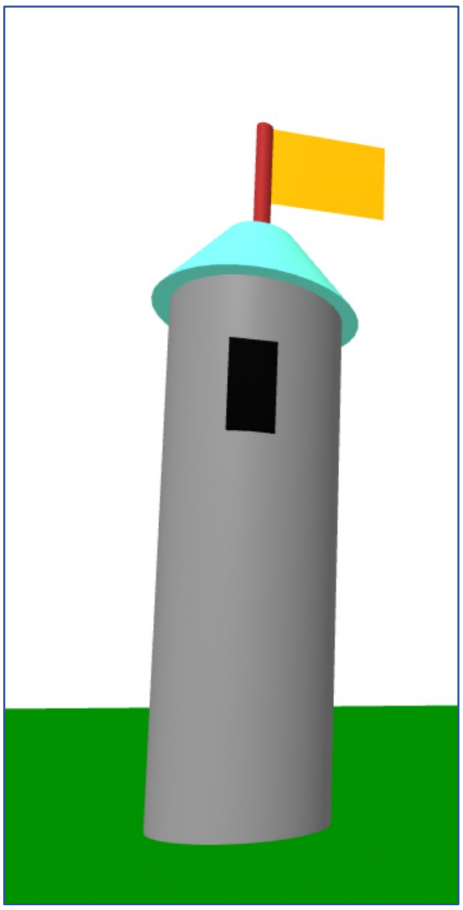
- `Brown color`
- `Position -2 7 -6`
- `Radius 0.1`
- `Height 3`

Next we will add a plane as the actual flag. We will use a plane rather than a box as we don't want it to have any thickness.

- `Position -1.25 8 -6`
- `Width 1.5`
- `Height 1`
- `Rotation 0 0 0 (we want this one to be vertical, unlike the floor)`
- `You can set your own color using a hex code.`

Have a practice: Change the height of the flagpole and line up the position of the flag so it's still at the top.

3.5 Stage one review



Our tower should look like the one on the right, perhaps with different colors that you have selected.

- Do you see the similarity to using different shape tags? They all have the same format
- Most share some properties like position, but different shapes will have their own unique properties too

Before we move on, make sure you are comfortable with how each of the shapes works, and the coordinates in particular, as we will be doing some more complicated work on these soon!

Before you continue: Fill out the code you have learnt on the last page reference guide

3.6 Castle Front Wall

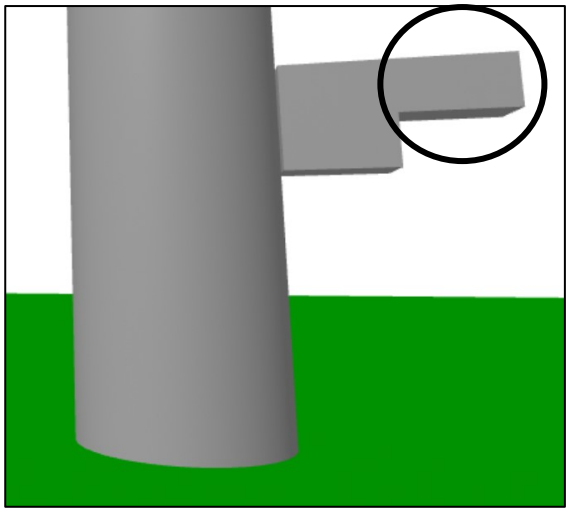
The front castle wall is made up of lots of different boxes.

This is for two reasons: 1) It gives you chance to further practice and master using coordinates, and secondly, when we add the textures later, they look better wrapped around smaller blocks.

We will start with the upper wall about the draw bridge opening, which will use four boxes and will join our front two towers.

- `Create a box with the following properties:`
- `Grey color`
- `Position -0.5 3 -6`
- `Width 1`
- `Height 1`
- `Depth 0.5`

We will now add the second box, but it will step up to make room for the drawbridge that we will add later.

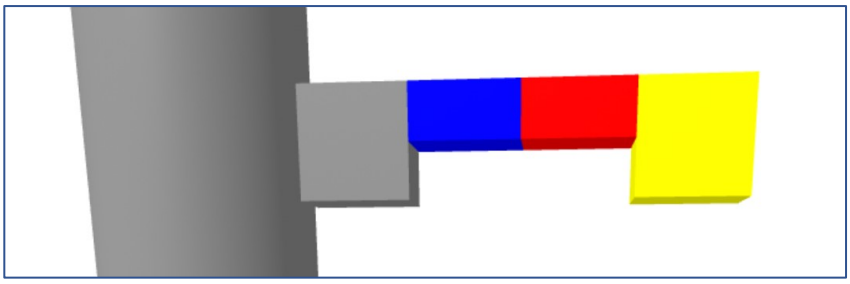


The dimensions are the same as the first block, however it is only half the height.

- `Position 0.5 3.25 -6`
- `Height 0.5`

Notice the changes in position. We move right by the width of one block (width of 1), which moves our position from -0.5 to 0.5.

The picture to the right shows what the completed top wall will look like. Each block is shown in different color to make it easier to see how each one fits.



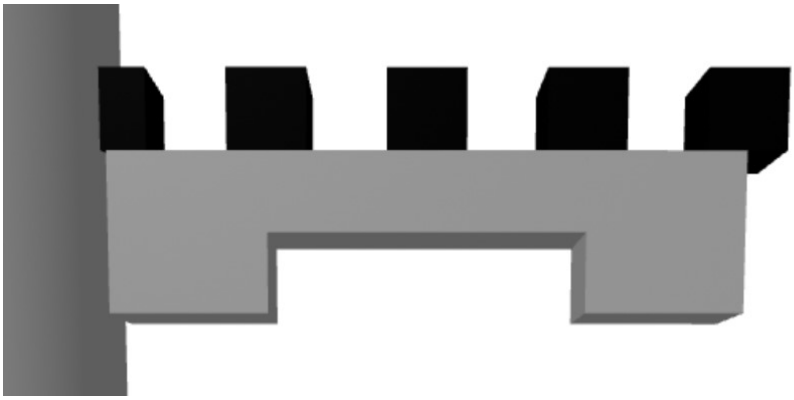
Work out the position of the red and yellow blocks above in this table and add them to your project:

First Grey Block	Blue block	Red block	Yellow Block
`-0.5 3 -6`	`0.5 3.25 -6`		

3.7 Adding Battlements

The battlements, shown here in black are a bit easier as they are all made from a box, are the same size, and evenly spaced apart.

Notice that the end blocks actually merge into the towers.



Here are the properties for the first block:

- Grey color
- Position -1 3.75 -6
- Width, height and depth all 0.5

Each block is a distance of 1 apart. Complete the table below and then add your battlements:

Battlement 1	Battlement 2	Battlement 3	Battlement 4	Battlement 5
-1 3.75 -6				

3.8 Front wall left and right sides of drawbridge opening

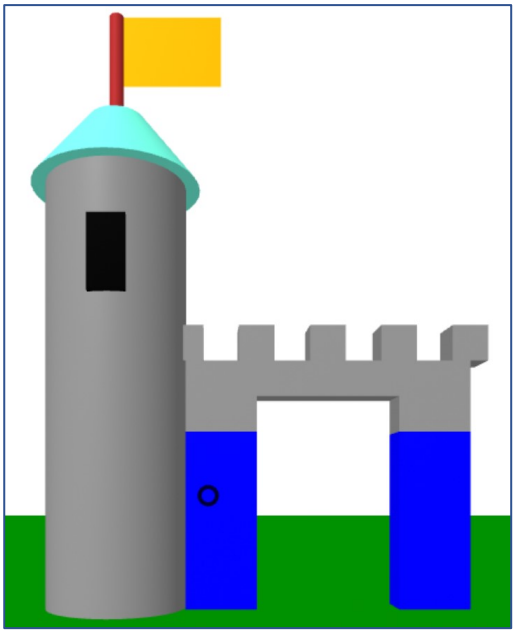
The final part of our front wall are the side parts, shown here in blue.

First, let's make the left wall.

Create a box with the following properties:

- Grey color
- Position -0.5 1 -6
- Width 1
- Height 3
- Depth 0.5

The wall to the right is a distance of 2 from the left. Have a try of adding that one yourself.



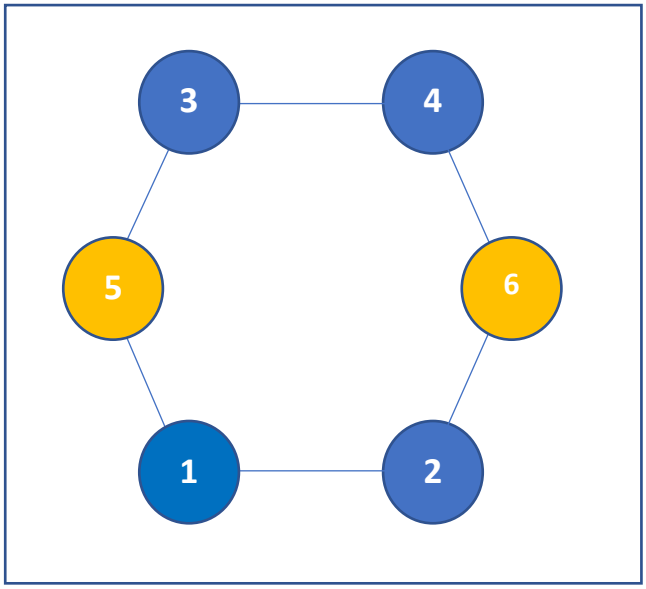
3.9 Adding the other towers

Our castle has a total of 6 towers.

The diagram to the right shows **a top view** of the castle, with **number one** the front left tower we have already completed.

All the blue towers are the **same height**, and have a cone roof on them

The orange towers are **shorter** without a cone roof on top.



Using the following information, **fill out the table below with the positions of each tower**:

- Tower 2 (Front right tower) is a distance of 6 to the right from tower 1
- The distance between towers 3 and 4 is the same as between towers 1 and 2
- Tower 3 is a distance of 10 further back from tower 1
- The distance between towers 1 and 3 is the same as towers 2 and 4
- Towers 5 and 6 are 1 unit shorter than the others
- Towers 5 and 6 are halfway between the front (1 and 2) and rear (3 and 4) towers
- The distance between towers 5 and 6 is 10

	Tower 1	Tower 2	Tower 3	Tower 4	Tower 5	Tower 6
X	-2					
Y	3					
Z	-6					

Now use a similar system to work out the Cone roofs and Windows on Towers 1 to 4.

	Tower 1	Tower 2	Tower 3	Tower 4
Roof X **Window X**	-2 -2			
Roof Y **Window Y**	6.5 4.75			
Roof Z **Window Z**	-6 -5			

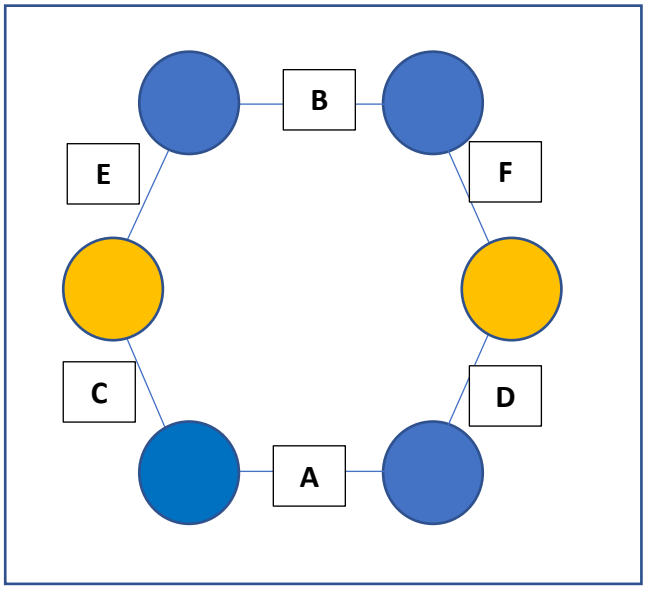
3.10 Adding walls between the towers

Our castle needs 5 more walls

The diagram to the right shows **a top view** of the castle, with **A** the front wall we have already completed.

Wall B is the rear wall of the castle, and walls C, D, E and F make up the sides.

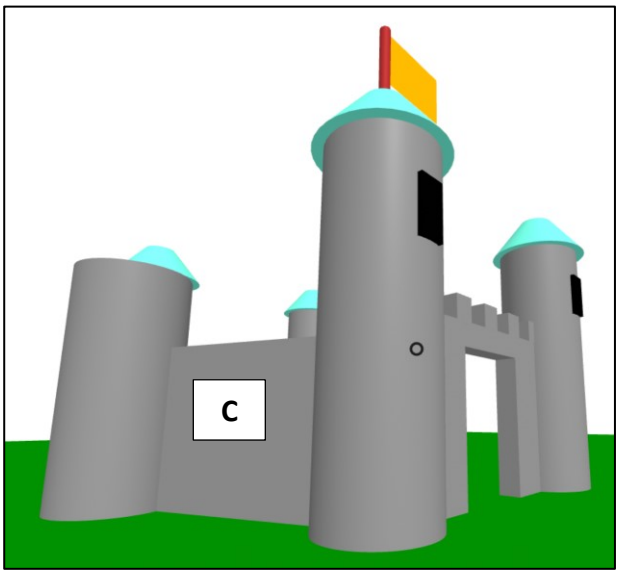
We will need to rotate the walls now, as well as position them, but we can use a similar system as the towers. As long as you know one angle, and one postion, you can use a table to work out the others.



Let's start with the C wall, on the front left-hand side:

Start by making a box, with the following properties:

- `Grey color`
- `Position -2.5 1 -8`
- `Rotation 0 -70 0`
- `Width 5`
- `Height 4`
- `Depth 0.5`



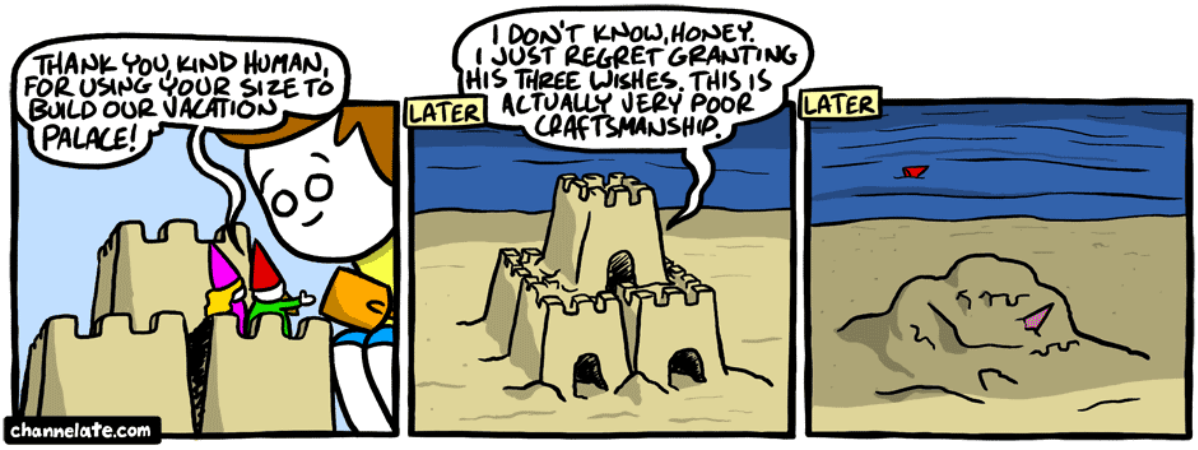
Now it's starting to look more like a castle!

Now it's your turn to create the rest of the walls.

You already know the position of the towers, so this should help you position your walls. Also think carefully about the rotation of the walls as they are all related!

	Wall C	Wall D	Wall E	Wall F	Wall B
Rotation					
Position	0 -70 0				
X	-2.5				
Y	1				
Z	-8				

Hopefully your castle walls are more successful than the ones below 😂



Challenge Time!

Now that your castle is starting to take shape, it's time to add your own personal touch!

Challenge 1 (Easy)

- Add battlements to your side castle walls

Challenge 2 (Medium)

- Add a castle Keep between the rear towers. You can join it to the back wall if you like
- Don't forget to add windows and an entry door from the bailey (courtyard)

Challenge 3 (Hard)

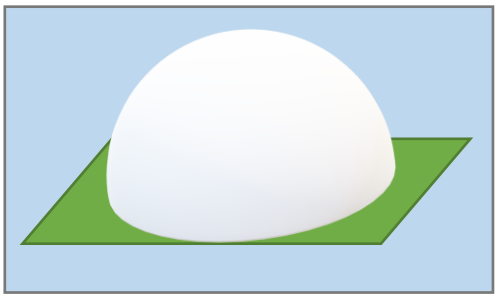
- Turn one of the middle towers into a round fortified tower top, complete with battlements.

You can refer to page 3 for some inspiration for these challenges!

4. Adding Finishing Touches

Now that we have the basic shape of our castle, it's time to add some details to make it look more realistic. Let's start with some textures.

4.1 Adding a sky



The sky tag wraps a sky around a dome that covers your entire plane, similar to the image to the right.

By wrapping the image, we get a more realistic looking sky, no matter the angle we view it.

Add the following tag to the Sky section of your code:

```
<a-sky> </a-sky>
```

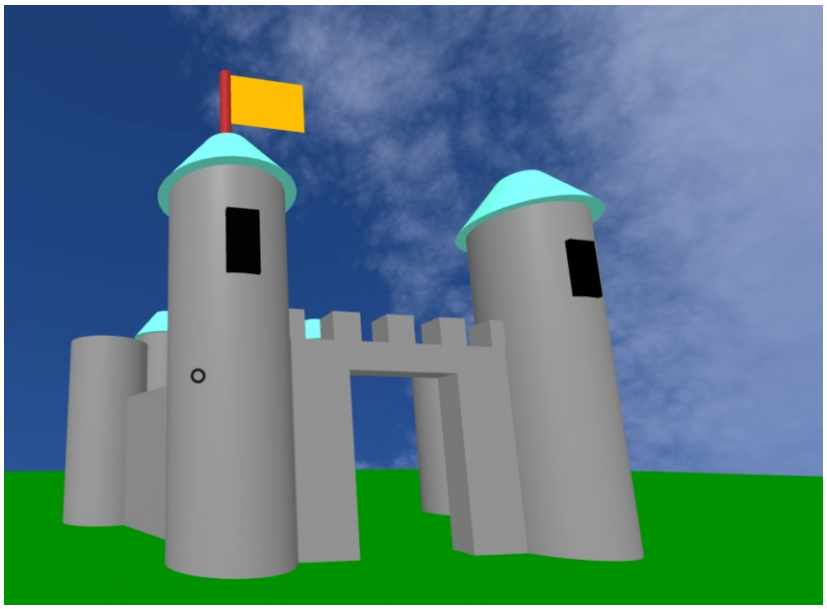
You can add a color property just like any other shape, but we want to add something that looks better using a texture (pre-made graphic).

- Download the file sky.jpg from the link below and add it to your work folder.
  - www.ai3.academy/inspire/metaverse/sky.jpg
- Once this is done and you can see it in your workspace, add the following property

```
src="/sky.jpg"
```

You should now have a realistic sky with clouds over your castle!

Look around your world and see how the sky is all around you!



4.2 Castle Textures

Now we will do the same process to add textures to most of the castle elements.

You can download the texture files from:

www.ai3.academy/inspire/metaverse

Make sure to place them all in the same folder as your html file, and check they appear in your workspace.

Let's have a look at what the tag and properties for the front left tower looks like with the texture added:

```
<a-cylinder color="grey" position="-2 3 -6"
            width="2" height="6" depth="2" src="/bricks4p.jpeg">
</a-cylinder>
```

- Note that in the above example we have a color AND texture property.
- You can change the color with a texture to give it a 'tint'. Give it a try!

This table shows you which texture is recommended for which element:

Towers	bricks4p.jpeg
Side Walls	bricks2p.jpg
Battlements	brick1p.jpg
Small Front wall blocks	bricks1p.jpg
Front Wall Sides	bricks3p.jpg"
Grass	grass3.jpg
	grass2.jpg
	Grass1.jpg
Bailey	castle-floor.jpg
Moat	water-a.gif
	Water.jpg

- To download the above images, in your browser, type the following path:

www.ai3.academy/inspire/metaverse/*filename*

- eg: for the towers texture, you would type:

www.ai3.academy/inspire/metaverse/bricks4p.jpeg

- Once the image is displayed, RIGHT click it and select ***Save image as***, and add it to your workspace folder.

4.3 Castle Bailey (Courtyard)

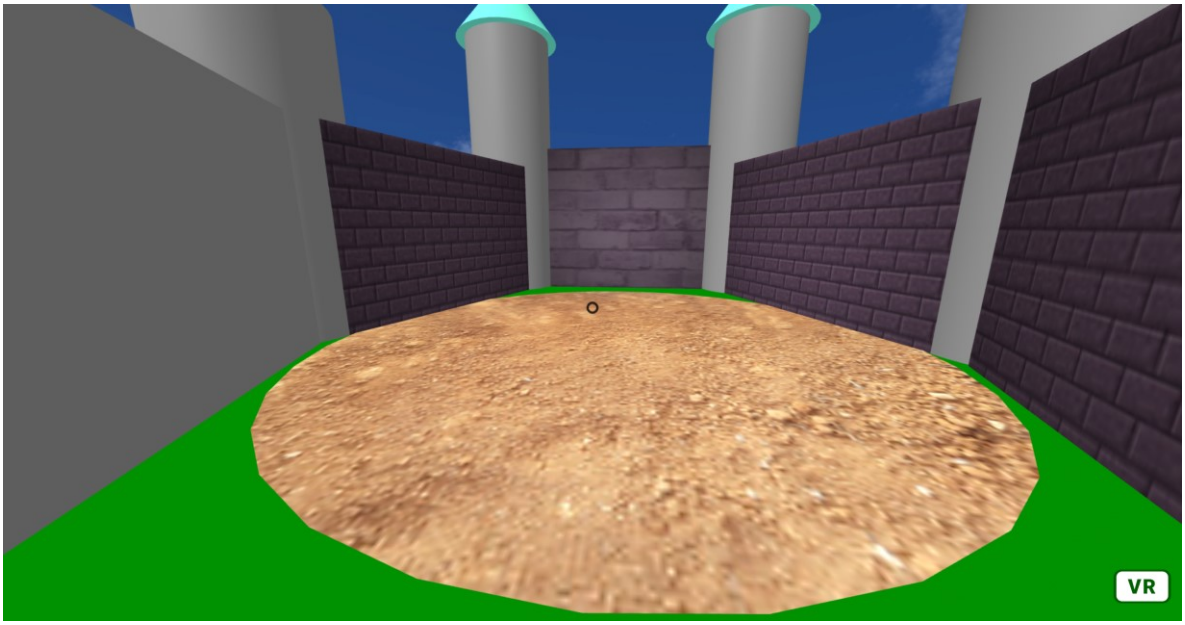
Now we want to add a bailey, or courtyard. We will use a new shape, a circle. It is much the same as a plane (remember the rotation!), but it's round!

```
<a-circle>    </a-circle>
```

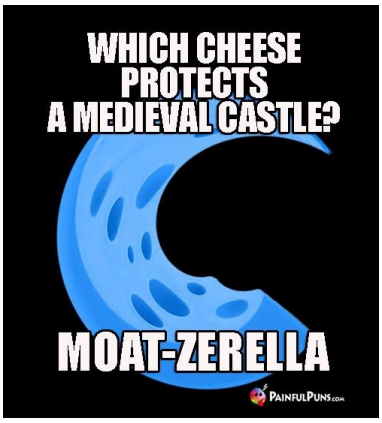
Using the following properties, see if you can place the bailey exactly as shown in the image below.

- `Radius 4`
- `Height 0.2`
- `Depth 2`

Once you are done you can add the texture.



4.4 Castle Moat



Every good castle needs a moat for protection! Hopefully yours is a bit better than the moat joke here…. (it's a bit cheesy)

To create our moat, we will learn a new tag – Ring.

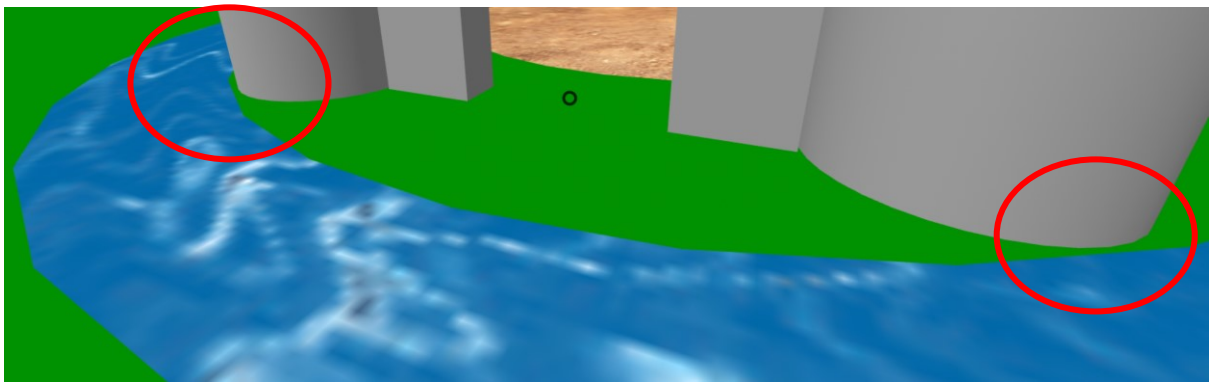
```
<a-ring>  </a-ring>
```

If you got the bailey correct, the moat will use exactly the same position and rotation.

Here are the rest of the properties:

- `radius-inner 7`
- `radius-outer 8.4`
- `height 0.2`
- `depth 2`

If you position your moat correctly, you will have just a tiny gap from each of the front and back towers.



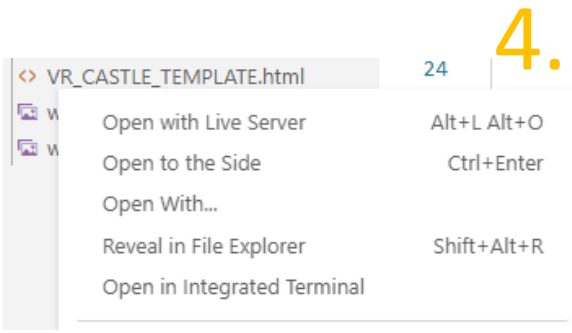
One more thing to flag…

- One last texture to add….. Design your own, or find a texture for your flag on top of your tower online.

5. Testing your world on a VR headset

1. **Build** your Cardboard VR Headset

2. Make a sure that your PC & Mobile are connected through same network.
3. Disconnect your Mobile roaming data
4. Open your code in live server
5. Take a note of the PORT number. In the example it is 5500
6. Find your IP Address as follows:



- **Windows** : Open `CMD` and enter `ipconfig`.
- **Linux/macOS** : Open `terminal` and enter `ifconfig`.

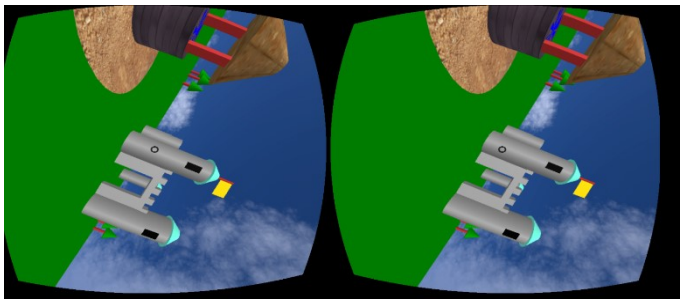
7. In your phone browser, enter the following:

http://your ip address:port number

For example: http://192.168.0.0.1:5500

8. You should see your VR world in your phone browser now!

   Select the VR button in the bottom right corner of the screen, and then put your phone into the cardboard headset.



Challenge 4 - Village Well

- Can you work it out from the picture?
- Hint: Position the Well at: `-5 0.5 0`

Here are the elements it needs:

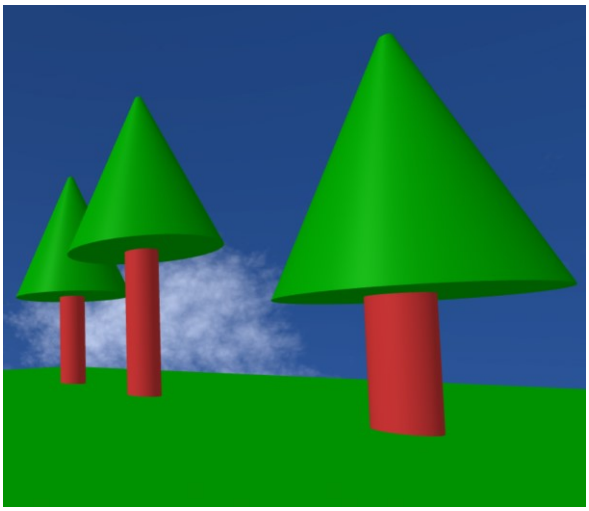
1. Well
2. Roof
3. Posts
4. Ground under well
5. Water

Challenge 5 – Tree

- Can you work it out from the picture?
- Hint: Position one tree at: `32 1 -35`

Here are the elements it needs:

1. Branches (main body of tree)
2. Trunk



Can you turn you tree into a forest using different size trees, or different designs?

END OF PART ONE

APPENDIX A: Dragon Realm VR – A-Frame Quick Reference guide

Item	Tag Syntax
Comment	<!-- *comment text goes here* -->
Floor (Plane)	
Cylinder	
Box	
Cone	
Properties	
Properties	
Sky	
Textures	
Circle	
Ring	